**Database Development and Class Registration**

Tony Carolla

The University of Arizona Global Campus

CST 499 Capstone for Computer Software Technology

Amr Elchouemi

January 23, 2023

**Database Development and Class Registration**

The MetricuL8 Class Registration system is ready for acceptance testing. We have added prescribed functionality, including the maintenance of classes and student enrollments. We hope you find that all functionality meets or exceeds your expectations.

**Figure 1**

*Maintain Classes*

Graphical user interface, website

Description automatically generated

**Figure 2**

*Maintain Enrollments*

Graphical user interface

Description automatically generated

**Figure 3**

*Database Creation*

CREATE

DATABASE MetricuL8v2;

USE MetricuL8v2;

CREATE

TABLE `student`

(`id` int(11) NOT NULL COMMENT 'Student ID',

`email` varchar(1000) NOT NULL COMMENT 'Student Email',

`password` varchar(256) NOT NULL COMMENT 'Password',

`firstName` varchar(100) NOT NULL COMMENT 'First Name',

`lastName` varchar(100) NOT NULL COMMENT 'Last Name',

`phoneArea` char(3) NOT NULL COMMENT 'Phone Area Code',

`phoneNumber` char(7) NOT NULL COMMENT 'Phone Number',

`birthDate` date NOT NULL COMMENT 'Date of Birth',

PRIMARY KEY (`id`),

CONSTRAINT uq\_email UNIQUE(`email`)

) COMMENT='Student Table';

INSERT INTO student (`id`, `email`, `password`, `firstname`, `lastname`, `phonearea`, `phonenumber`, `birthdate`)

VALUES(0, 'admin@metricul8.com', 'admin99999', 'admin', 'blank', 888, 5551212, '1900-01-01');

CREATE

TABLE `class`

(`id` int(11) not null comment 'Class ID',

`classDescription` varchar(100) not null comment 'Class Description',

`year` int(11) not null comment 'Class Year',

`semester` int(11) not null comment 'Class Semester',

`maxEnrollment` int(11) not null comment 'Maximum Class Enrollment',

PRIMARY KEY (`id`, `year`, `semester`)

) COMMENT='Class Table' ;

INSERT INTO class (`id`, `classDescription`, `year`, `semester`, `maxEnrollment`)

VALUES(0, 'adminClass', 2000, 1, 1);

CREATE

TABLE `enrollment`

(`id` int(11) not null comment 'Enrollment ID',

`studentId` int(11) not null comment 'Student ID',

`classId` int(11) not null comment 'Class ID',

PRIMARY KEY (`id`),

CONSTRAINT uq\_student\_class UNIQUE(`studentId`, `classId`),

CONSTRAINT fk\_enrollment\_class FOREIGN KEY (`classId`) REFERENCES `class`(`id`),

CONSTRAINT fk\_enrommnent\_student FOREIGN KEY (`studentId`) REFERENCES `student`(`id`)

) COMMENT='Enrollment Table' ;

INSERT INTO enrollment (`id`, `studentId`, `classId`)

VALUES(0, 0, 0) ;

CREATE

TABLE `waitList`

(`id` int(11) not null comment 'Wait List ID',

`classId` int(11) not null comment 'Class ID',

`studentId` int(11) not null comment 'Student ID',

`waitingSince` timestamp not null comment 'Waiting Since',

PRIMARY KEY (`id`),

CONSTRAINT uq\_class\_student UNIQUE(`classId`,`studentId`),

CONSTRAINT fk\_waitList\_class FOREIGN KEY (`classId`) REFERENCES `class`(`id`),

CONSTRAINT fk\_waitlist\_student FOREIGN KEY (`studentId`) REFERENCES `student`(`id`)) ;

INSERT INTO waitList (`id`, `classId`, `studentId`, `waitingSince`)

VALUES(0, 0, 0, CURRENT\_TIMESTAMP()) ;

DELIMITER //

CREATE

PROCEDURE enrollStudent (IN classId INT, IN studentId INT)

BEGIN

DECLARE cSize,

maxSize,

nextId

INT DEFAULT 0;

SELECT cSize = COUNT(\*)

FROM enrollment

WHERE enrollment.classId = classId;

SELECT maxSize = class.maxEnrollment

FROM class

WHERE class.id = classId;

IF ( cSize <= maxSize ) THEN

SELECT nextId = COUNT(\*)

FROM enrollment;

IF ( nextId = 0 ) THEN

SET nextId = 1;

ELSE

SET nextId = 0;

SELECT nextId = MAX(id)+1

FROM enrollment;

END IF;

INSERT

INTO enrollment

VALUES (nextId, studentId, classId);

SELECT 'A' AS addResult;

ELSE

SET cSize = 0;

SELECT cSize = COUNT(\*)

FROM waitlist

WHERE waitlist.classId = classId

AND waitlist.studentId = studentId;

IF (cSize = 0) THEN

SET nextId = 0;

SELECT nextId = COUNT(\*)

FROM waitlist;

IF ( nextId = 0 ) THEN

SET nextId = 1;

ELSE

SET nextId = 0;

SELECT nextId = MAX(id)+1

FROM waitlist;

END IF;

INSERT

INTO waitlist

VALUES (nextId, classId, studentId, CURRENT\_TIMESTAMP());

END IF;

SELECT 'W' AS addResult;

END IF;

END//

CREATE

PROCEDURE removeStudent (IN enrollmentId INT)

BEGIN

DELETE FROM enrollment WHERE id = enrollmentId;

END//

DELIMITER ;

**Figure 4**

*Main.css*

#jt {

font-family: "Franklin Gothic Medium", serif;

background-image: url("../img/OceanGrid.png");

background-repeat: repeat;

color: #0C2E7E;

}

#addText {

text-align: center;

}

#epNavBar {

font-family: "Arial Narrow", sans-serif;

}

#homeIcon {

font-size: 64px;

}

#availSel option{

width:300px;

}

#enrollSel option{

width:300px;

}

.AddressLong {

width: 40%;

}

.bgBody {

background-color: #e1fafd;

}

.ButtonFont {

font-size: 32px;

}

.ButtonText {

font-weight: bold;

}

.error {

color: red;

}

.FloatInput {

margin-bottom: 15px;

}

.ItemName {

font-weight: bold;

}

.ListLabel {

font-style: italic;

color: #0C2E7E;

}

.StateShort {

width: 60%;

}

.ZipMedium {

width: 60%;

}

.PhoneLarge {

width: 80%;

}

.PhoneMedium {

width: 90%;

}

**Figure 5**

*Classes.php*

<?php

error\_reporting(E\_ALL ^ E\_NOTICE);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title> Classes Page </title>

<meta charset="utf8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="stylesheet" href="css/normalize.css">

<link rel="stylesheet" href="css/main.css">

<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

</head>

<body class="bgBody">

<?php require 'master.php';?>

<?php

if ($\_SERVER["REQUEST\_METHOD"]=="POST") {

$anyErr = false;

$errorsExist = "";

$deleteError = "";

$descriptionError = "";

$yearError = "";

$semesterError = "";

$maxEnrollmentError = "";

$addError = "";

// set from input form

if (!empty($\_POST["description"]))

{

$description = $\_POST["description"];

}

if (!empty($\_POST["year"]))

{

$year = $\_POST["year"];

}

if (!empty($\_POST["semester"]))

{

$semester = $\_POST["semester"];

}

if (!empty($\_POST["maxEnrollment"]))

{

$maxEnrollment = $\_POST["maxEnrollment"];

}

if (isset($\_POST["Delete"]) && is\_null($\_POST['allClass']))

{

$anyErr = true;

$deleteError = "Choose Class";

}

if (!isset($\_POST["Delete"]))

{

if (empty($\_POST["description"])) {

$anyErr = true;

$descriptionError = "Description required";

}

if (empty($\_POST["year"])) {

$anyErr = true;

$yearError = "Year required";

}

if (empty($\_POST["semester"])) {

$anyErr = true;

$semesterError = "Semester required";

}

if (empty($\_POST["maxEnrollment"])) {

$anyErr = true;

$maxEnrollmentError = "Max Enrollment required";

}

}

if ($anyErr)

{

$errorsExist = "Fix the errors below";

} else {

$dbStore = new MetricuL8Store();

if (isset($\_POST["Delete"]))

{

foreach ($\_POST['allClass'] as $theClass)

{

$dbStore->cancelClass($theClass);

}

$errorsExist = "Class(es) Removed.";

} else {

$dbStore->addclass($description, $year, $semester, $maxEnrollment);

$errorsExist = "Class Added!";

}

$description = "";

$year = "";

$semester = "";

$maxEnrollment = "";

$deleteError = "";

$descriptionError = "";

$yearError = "";

$semesterError = "";

$maxEnrollmentError = "";

$addError = "";

}

}

// generate list of classes into $optionList

$dbStore2 = new MetricuL8Store();

$selectedClasses = array();

if ($\_SERVER["REQUEST\_METHOD"]=="POST"

&& !is\_null($\_POST['allClass']))

{

$selectedClasses = $\_POST['allClass'];

}

if($cResult = $dbStore2->getClassList()){

foreach ($cResult as $row){

if (in\_array($row['id'], $selectedClasses))

{

$optionList .= '<option value="' . $row['id'] . '" selected="selected">' . $row['description'] . '</option>';

} else {

$optionList .= '<option value="' . $row['id'] . '">' . $row['description'] . '</option>';

}

}

} else {

$optionList = '';

}

?>

<div class="container">

<h1>New Class</h1>

<p class="error"><?php echo $errorsExist;?></p>

<form class="form-horizontal" role="form" method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

<div class="form-group">

<label class="control-label col-sm-2" for="f-description">Description</label>

<div class="col-sm-10">

<input type="text" name="description" id="f-description" value="<?php echo $description;?>" />

<span class="error"><?php echo $descriptionError; ?></span>

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="f-year">Year</label>

<div class="col-sm-10">

<input type="number" name="year" id="f-year" value="<?php echo $year;?>" />

<span class="error"><?php echo $yearError; ?></span>

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="f-semester">Semester</label>

<div class="col-sm-10">

<input type="number" name="semester" id="f-semester" value="<?php echo $semester;?>" />

<span class="error"><?php echo $semesterError; ?></span>

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="f-maxEnrollment">Max Enrollment</label>

<div class="col-sm-10">

<input type="number" name="maxEnrollment" id="f-maxEnrollment" value="<?php echo $maxEnrollment;?>" />

<span class="error"><?php echo $maxEnrollmentError; ?></span>

</div>

</div>

<div class="form-group">

<div class="col-sm-6">

<input type="submit" value="Add" name="Add" />

</div>

<div class="col-sm-6">

<span class="error"><?php echo $addError; ?></span>

</div>

</div>

<div class="col-sm-12">

<h1>Cancel Class</h1>

</div>

<div class="form-group">

<div class="col-sm-12">

<select name="allClass[]" multiple id="classSel">

<?php echo $optionList; ?>

</select>

</div>

</div>

<div class="col-sm-12">

<input type="submit" value="Delete" name="Delete" />

</div>

<div class="col-sm-12">

<span class="error"><?php echo $deleteError; ?></span>

</div>

</form>

</div>

<?php require\_once 'footer.php';?>

</body>

</html>

**Figure 6**

*Enrollments.php*

<?php

error\_reporting(E\_ALL ^ E\_NOTICE);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title> Enrollments Page </title>

<meta charset="utf8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="stylesheet" href="css/normalize.css">

<link rel="stylesheet" href="css/main.css">

<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

</head>

<body class="bgBody">

<?php require 'master.php';?>

<?php

if ($\_SERVER["REQUEST\_METHOD"]=="POST") {

$anyErr = false;

$removeError = "";

$addError = "";

if (isset($\_POST["Add"]) && is\_null($\_POST['allAvailable']))

{

$anyErr = true;

$addError = "Choose Class";

}

if (isset($\_POST["Remove"]) && is\_null($\_POST['allEnrolled']))

{

$anyErr = true;

$removeError = "Choose Class";

}

if (!$anyErr)

{

$dbStore = new MetricuL8Store();

if (isset($\_POST['Remove'])) {

$enrollmentId = $\_POST['allEnrolled'];

$dbStore->removeStudent($enrollmentId);

$removeError = 'Enrollment Removed';

}

else {

$studentId = $dbStore->getUserId($\_SESSION['username']);

$classId = $\_POST['allAvailable'];

$addResult = $dbStore->enrollStudent($classId, $studentId);

if ($addResult == 'W'){

$addError = '-- Placed on wait list --';

}

else {

$addError = 'Student Enrolled';

}

}

}

}

$dbStore2 = new MetricuL8Store();

$userId = $dbStore2->getUserId($\_SESSION['username']);

// generate list of enrolled classes

$selectedEnrollments = "";

if ($\_SERVER["REQUEST\_METHOD"]=="POST"

&& !is\_null($\_POST['allEnrolled'])) {

$selectedEnrollments = $\_POST['allEnrolled'];

}

if($cResult = $dbStore2->getEnrollments($userId)){

foreach ($cResult as $row){

if ($row['id'] == $selectedEnrollments)

{

$enrollOptionList .= '<option value="' . $row['id']

. '" selected="selected">' . $row['description']

. '</option>';

} else {

$enrollOptionList .= '<option value="' . $row['id']

. '">' . $row['description'] . '</option>';

}

}

} else {

$enrollOptionList = '';

}

$selectedAdds = "";

if ($\_SERVER["REQUEST\_METHOD"]=="POST"

&& !is\_null($\_POST['allAvailable'])) {

$selectedAdds = $\_POST['allAvailable'];

}

if($dResult = $dbStore2->getAvailable($userId)){

foreach ($dResult as $row){

if ($row['id'] == $selectedAdds)

{

$addOptionList .= '<option value="' . $row['id']

. '" selected="selected">' . $row['description']

. '</option>';

} else {

$addOptionList .= '<option value="' . $row['id']

. '">' . $row['description'] . '</option>';

}

}

} else {

$addOptionList = '';

}

?>

<div class="container">

<h1>Maintain Enrollments</h1>

<form class="form-horizontal" role="form" method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

<div class="form-group">

<div class="col-sm-5">

<select name="allEnrolled" id="enrollSel" size="5">

<option disabled>Enrolled Classes:</option>

<?php echo $enrollOptionList; ?>

</select>

</div>

<div class="col-sm-2">

<input type="submit" value="Add" name="Add" />

</div>

<div class="form-group">

<div class="col-sm-5">

<select name="allAvailable" id="availSel" size="5">

<option disabled>Available Classes:</option>

<?php echo $addOptionList; ?>

</select>

</div>

</div>

</div>

<div class="form-group">

<div class="col-sm-5">

&nbsp;

</div>

<div class="col-sm-2">

<input type="submit" value="Remove" name="Remove" />

</div>

<div class="col-sm-5">

&nbsp;

</div>

</div>

<div class="form-group">

<div class="col-sm-6">

<span class="error"><?php echo $removeError; ?></span>

</div>

<div class="col-sm-6">

<span class="error"><?php echo $addError; ?></span>

</div>

</div>

</form>

</div>

<?php require\_once 'footer.php';?>

</body>

</html>

**Figure 7**

*Footer.php*

<?php

error\_reporting(E\_ALL ^ E\_NOTICE)

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

</head>

<body>

<div class="navbar-fixed-bottom row-fluid">

<div class="navbar-inner">

<div class="container text-center">

Copyright @ 2023

</div>

</div>

</div>

</body>

</html>

**Figure 8**

*Index.php*

<?php

error\_reporting(E\_ALL ^ E\_NOTICE);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title> Home Page </title>

<meta charset="utf8" />

<meta name="description" content="" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="stylesheet" href="css/main.css">

<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

</head>

<body class="bgBody">

<?php

if (session\_id() == "") {

ini\_set('session.use\_only\_cookies','1');

session\_start();

}

if(isset($\_GET['Logout']) && $\_GET['Logout']=='1') {

unset($\_SESSION['username']);

}

?>

<?php require 'master.php';?>

<div class="container">

<div class="row">

<div class="col-md-3" id="homeIcon">

<span class="glyphicon glyphicon-user"></span>

</div>

<div class="col-md-9">

<h2>Welcome to the MetricuL8 Student Enrollment System!</h2>

<p>Use the links above to access your enrollment records. </p>

</div>

</div>

</div>

<?php require\_once 'footer.php';?>

</body>

</html>

**Figure 9**

*Login.php*

<?php

error\_reporting(E\_ALL ^ E\_NOTICE);

require\_once 'MetricuL8Store.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title> Login Page </title>

<meta charset="utf8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="stylesheet" href="css/main.css">

<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

</head>

<body class="bgBody">

<?php

if (session\_id() == "") {

ini\_set('session.use\_only\_cookies','1');

session\_start();

}

if( !isset($\_SESSION['username']) && $\_SERVER['REQUEST\_METHOD']=='POST' ) {

$dbStore = new MetricuL8Store();

if( $dbStore->checkCred($\_POST['usernameTry'], $\_POST['passKey']) ) {

$\_SESSION['username'] = $\_POST['usernameTry'];

if(isset($\_SESSION['upError'])){

unset($\_SESSION['upError']);

}

} else {

$\_SESSION['upError'] = true;

}

}

?>

<?php require 'master.php';?>

<?php

if (session\_id() == "") {

ini\_set('session.use\_only\_cookies','1');

session\_start();

}

if (isset($\_SESSION['username'])) {

echo makeWelcomeForm($\_SESSION['username']);

} else {

echo makeLoginForm();

}

function makeWelcomeForm($uName){

$welcome = '<div class="container"><p>Welcome '. $uName .'</p></div>';

return $welcome;

}

function makeLoginForm() {

if(isset($\_POST['usernameTry'])) {

$usernameTry = $\_POST['usernameTry'];

$passKey = $\_POST['passKey'];

} else{

$usernameTry='';

$passKey='';

}

$html = '<div class="container">';

$html .= '<h1>Log in to MetricuL8</h1>';

if (isset($\_SESSION["upError"])) {

$html .= '<span class="error">Error unable to log in.</span>';

unset($\_SESSION['upError']);

}

$html .='<form class="form-horizontal" role="form" method="post" action="'

. htmlspecialchars($\_SERVER["PHP\_SELF"]) .'">';

$html .= '<div class="form-group">'

.'<label class="control-label col-sm-2" for="f-usernameTry">User Email</label>'

.'<div class="col-sm-10">'

.'<input type="email" name="usernameTry" id="f-usernameTry" placeholder="email@provider.com" value="'

.$usernameTry

.'" /></div></div>';

$html .= '<div class="form-group">'

.'<label class="control-label col-sm-2" for="f-password">Password</label>'

.'<div class="col-sm-10">'

.'<input type="password" name="passKey" id="f-password" value="'.$passKey.'" />'

.'</div></div>';

$html .= '<div class="form=group">'

.'<div class="col-sm-5 col-sm-offset-4">'

.'<button type="submit" title="Login" class="ButtonFont">'

.'<span class="glyphicon glyphicon-ok"></span>'

.'</button></div></div>'

.'</form></div>';

return $html;

}

?>

<?php require\_once 'footer.php';?>

</body>

</html>

**Figure 10**

*Master.php*

<?php

error\_reporting(E\_ALL ^ E\_NOTICE);

if (session\_id() == "") {

ini\_set('session.use\_only\_cookies','1');

session\_start();

}

if(isset($\_SESSION['username']))

echo "Welcome: " . $\_SESSION['username'];

require\_once 'MetricuL8Store.php';

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8" />

<meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1"/>

<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

</head>

<body>

<div class="jumbotron" id="jt">

<div class="containter text-center">

<h1>MetricuL8</h1>

<h2>Student Enrollment System</h2>

</div>

</div>

<nav class="navbar navbar-inverse">

<div class="containter-fluid">

<div class="navbar-header">

<button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#epNavBar">

<span class="icon-bar"></span>

<span class="icon-bar"></span>

<span class="icon-bar"></span>

</button>

</div>

<div class="collapse navbar-collapse" id="epNavBar">

<ul class="nav navbar-nav">

<li class="active"><a href="index.php"><span class="glyphicon glyphicon-home"></span> Home</a></li>

</ul>

<ul class="nav navbar-nav navbar-right">

<?php

if (session\_id() == "")

{

ini\_set('session.use\_only\_cookies','1');

session\_start();

}

if(isset($\_SESSION['username']))

{

echo "<li><a href=\"enrollments.php\"><span class=\"glyphicon glyphicon-tags\"></span>&nbspEnrollments</a></li>";

if ($\_SESSION['username']=="admin@metricul8.com")

{

echo "<li><a href=\"classes.php\"><span class=\"glyphicon glyphicon-calendar\"></span>Classes</a></li>";

}

echo "<li><a href=\"index.php?Logout=1\"><span class=\"glyphicon glyphicon-off\"></span>Logout</a></li>";

}

else

{

echo "<li><a href=\"login.php\"><span class=\"glyphicon glyphicon-user\"></span> Login</a></li>";

echo "<li><a href=\"registration.php\"><span class=\"glyphicon glyphicon-pencil\"></span> Registration</a></li>";

}

?>

</ul>

</div>

</div>

</nav>

</body>

</html>

**Figure 11**

*MetricuL8Store.php*

<?php

class MetricuL8Store {

function removeStudent($enrollmentId){

$con='mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$';

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$pdo->query( "CALL removeStudent (" . $enrollmentId . ")" );

$pdo = null;

}

function enrollStudent($classId, $studentId) {

try {

$con='mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$';

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$sql = "CALL enrollStudent ("

.$classId . ", " . $studentId

.")";

$result = $pdo->query($sql);

$row = $result->fetch();

$addResult = $row['addResult'];

$pdo = null;

return $addResult;

}

catch (PDOException $e) {

die( $e->getMessage() );

}

}

function getUserId($email) {

$con='mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$';

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$sql = "SELECT MAX(id) AS id FROM student WHERE email='"

.$email

."'";

$result = $pdo->query($sql);

$row = $result->fetch();

$id = $row['id'];

$pdo = null;

return $id;

}

function getAvailable($studentId) {

try{

$con='mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$';

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$sql = "SELECT class.Id AS id, concat\_ws('-', class.year, class.semester, class.classDescription) AS description FROM class LEFT JOIN enrollment ON class.id = enrollment.classId AND enrollment.studentId = "

. $studentId

." WHERE enrollment.id IS NULL";

$result = $pdo->query($sql);

$resultSet = array();

while($row = $result->fetch()) {

$resultSet[] = $row;

}

$pdo = null;

return $resultSet;

}

catch(PDOException $e){

die( $e->getMessage() );

}

}

function getEnrollments($studentId) {

try{

$con='mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$';

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$sql = "SELECT enrollment.Id AS id, concat\_ws('-', class.year, class.semester, class.classDescription) AS description FROM class JOIN enrollment ON class.id=enrollment.classId WHERE enrollment.studentId = " . $studentId;

$result = $pdo->query($sql);

$resultSet = array();

while($row = $result->fetch()) {

$resultSet[] = $row;

}

$pdo = null;

return $resultSet;

}

catch(PDOException $e) {

die( $e->getMessage() );

}

}

function addclass($description, $year, $semester, $maxEnrollment) {

try {

$con='mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$';

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$result = $pdo->query('SELECT MAX(id)+1 AS maxid FROM class');

while($row = $result->fetch()) {

$nextId = $row['maxid'];

}

$stmt = $pdo->prepare('INSERT INTO class VALUES( :nextId, :description, :year, :semester, :maxEnrollment)');

$stmt->execute([ 'nextId' => $nextId, 'description' => $description, 'year' => $year, 'semester' => $semester, 'maxEnrollment' => $maxEnrollment ]);

$pdo = null;

return;

}

catch (PDOException $e) {

die( $e->getMessage() );

}

}

function cancelClass($id) {

try {

$con='mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$';

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$result = $pdo->query('SELECT COUNT(\*) AS rowCount FROM enrollment WHERE classid='.$id);

while($row = $result->fetch()) {

$rowCount1 = $row['rowCount'];

}

$result = $pdo->query('SELECT COUNT(\*) AS rowCount FROM waitlist WHERE classid='.$id);

while($row = $result->fetch()) {

$rowCount2 = $row['rowCount'];

}

if($rowCount1 == 0 && $rowCount2 == 0)

{

$sql = 'DELETE FROM class WHERE id='.$id;

$pdo->exec($sql);

}

$pdo = null;

return;

}

catch (PDOException $e) {

die( $e->getMessage() );

}

}

function getClassList() {

try {

$con='mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$';

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$result = $pdo->query("SELECT id, concat\_ws('-', year, semester, classDescription) AS description FROM class ORDER BY id");

$resultSet = array();

while($row = $result->fetch()) {

$resultSet[] = $row;

}

$pdo = null;

return $resultSet;

}

catch (PDOException $e) {

die( $e->getMessage() );

}

}

function executeSelectQuery($sql, $con="mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$") {

try {

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$result = $pdo->query($sql);

$resultSet = array();

while($row = $result->fetch()) {

$resultSet[] = $row;

}

$pdo = null;

return $resultSet;

}

catch (PDOException $e) {

die( $e->getMessage() );

}

}

function executeQuery($sql, $con="mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$") {

try {

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$count = $pdo->exec($sql);

$pdo = null;

return $count;

}

catch (PDOException $e) {

die( $e->getMessage() );

}

}

function checkCred($usernameTry, $passKeyTry) {

try {

$con='mysql:host=localhost;dbname=metricul8v2;user=EmeraldV;password=goraider$';

$pdo = new PDO($con);

$pdo->setAttribute(PDO::ATTR\_EMULATE\_PREPARES, false);

$pdo->setAttribute(PDO::ATTR\_ERRMODE, PDO::ERRMODE\_EXCEPTION);

$stmt = $pdo->prepare('SELECT COUNT(\*) AS stdcount FROM student WHERE email = :usernametry AND password= :passkeytry');

$stmt->execute([ 'usernametry' => $usernameTry, 'passkeytry' => $passKeyTry ]);

foreach($stmt as $row) {

$result = $row['stdcount'] == 1;

}

$pdo = null;

return $result;

}

catch (PDOException $e) {

die( $e->getMessage() );

}

}

}

?>

**Figure 12**

*Registration.php*

<?php

error\_reporting(E\_ALL ^ E\_NOTICE);

?>

<!DOCTYPE html>

<html lang="en">

<head>

<title> Registration Page </title>

<meta charset="utf8" />

<meta name="viewport" content="width=device-width, initial-scale=1" />

<link rel="stylesheet" href="css/normalize.css">

<link rel="stylesheet" href="css/main.css">

<link rel="stylesheet" href="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/css/bootstrap.min.css" />

<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.0/jquery.min.js"></script>

<script src="http://maxcdn.bootstrapcdn.com/bootstrap/3.3.6/js/bootstrap.min.js"></script>

</head>

<body class="bgBody">

<?php require 'master.php';?>

<?php

if ($\_SERVER["REQUEST\_METHOD"]=="POST") {

$anyErr = false;

$errorsExist = "";

$emailError = "";

$passKeyError = "";

$lastNameError = "";

$firstNameError = "";

$phoneError = "";

$dobError = "";

if (empty($\_POST["emailAddr"])) {

$anyErr = true;

$emailError = "Email required";

} else {

$emailAddr = $\_POST["emailAddr"];

}

if (empty($\_POST["passKey"])) {

$anyErr = true;

$passKeyError = "Password required";

} else {

$passKey = $\_POST["passKey"];

}

if (empty($\_POST["firstName"])) {

$anyErr = true;

$firstNameError = "First name required";

} else {

$firstName = $\_POST["firstName"];

}

if (empty($\_POST["lastName"])) {

$anyErr = true;

$lastNameError = "Last name required";

} else {

$lastName = $\_POST["lastName"];

}

if (empty($\_POST["birthDate"])) {

$anyErr = true;

$dobError = "Birth Date required";

} else {

$birthDate = $\_POST["birthDate"];

}

$phoneArea = $\_POST["phoneArea"];

$phoneNum = $\_POST["phoneNum"];

if ( empty($\_POST["phoneArea"]) || empty($\_POST["phoneNum"]) ) {

$anyErr = true;

$phoneError = "Phone# required";

}

if ($anyErr)

{

$errorsExist = "Fix the errors below";

} else {

// save to the database

$dbStore = new MetricuL8Store();

// get the max ID from the database

$sql = 'select max(id)+1 AS maxid from metricul8v2.student';

if($qResult = $dbStore->executeSelectQuery($sql)){

foreach ($qResult as $row){

$newId = $row['maxid'];

}

} else {

$newId = 1;

}

// build the insert statement

$sql = "insert into metricul8v2.student(id, email, password, firstName, "

. "lastName, phoneArea, phoneNumber, birthDate) values("

. strval($newId)

. ",'" . $\_POST["emailAddr"] . "'"

. ",'" . $\_POST["passKey"] . "'"

. ",'" . $\_POST["firstName"] . "'"

. ",'" . $\_POST["lastName"] . "'"

. "," . strval($\_POST["phoneArea"])

. "," . strval($\_POST["phoneNum"])

. ",'" . strval($\_POST["birthDate"]) . "'"

. ")";

if ($dbStore->executeQuery($sql)) {

$errorsExist = "Student Added!";

$emailAddr = "";

$passKey = "";

$lastName = "";

$firstName = "";

$phoneArea = 0;

$phoneNum = 0;

$birthDate = "";

$emailError = "";

$passKeyError = "";

$lastNameError = "";

$firstNameError = "";

$phoneError = "";

$dobError = "";

} else{

$errorsExist = "Can't save to the database. Contact your system administrator.";

}

}

}

?>

<div class="container">

<h1>New Student Registration</h1>

<p class="error"><?php echo $errorsExist;?></p>

<form class="form-horizontal" role="form" method="post" action="<?php echo htmlspecialchars($\_SERVER["PHP\_SELF"]);?>">

<div class="form-group">

<label class="control-label col-sm-2" for="f-email">Email Address</label>

<div class="col-sm-10">

<input type="email" name="emailAddr" id="f-email" placeholder="email@provider.com" value="<?php echo $emailAddr;?>" />

<span class="error"><?php echo $emailError; ?></span>

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="f-password">Password</label>

<div class="col-sm-10">

<input type="password" name="passKey" id="f-password" placeholder="8 or more characters" value="<?php echo $passKey;?>" />

<span class="error"><?php echo $passKeyError; ?></span>

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="f-firstName">First Name</label>

<div class="col-sm-10">

<input type="text" name="firstName" id="f-firstName" value="<?php echo $firstName;?>" />

<span class="error"><?php echo $firstNameError; ?></span>

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="f-lastName">Last Name</label>

<div class="col-sm-10">

<input type="text" name="lastName" id="f-lastName" value="<?php echo $lastName;?>" />

<span class="error"><?php echo $lastNameError; ?></span>

</div>

</div>

<div class="form-group">

<label class="control-label col-sm-2" for="f-birthDate">Date of Birth</label>

<div class="col-sm-10">

<input type="date" name="birthDate" id="f-birthDate" value="<?php echo $birthDate;?>" />

<span class="error"><?php echo $dobError; ?></span>

</div>

</div>

<div class="row show-grid FloatInput">

<div class="form=group">

<label class="control-label col-sm-2" for="f-phoneArea">Area Code/Phone#</label>

<div class="col-sm-1">

<input type="number" name="phoneArea" class="PhoneMedium" id="f-phoneArea" placeholder="###" value="<?php echo $phoneArea;?>" />

</div>

<div class="col-sm-2">

<input type="number" name="phoneNum" class="PhoneLarge" placeholder="#######" value="<?php echo $phoneNum;?>" />

</div>

<div class="col-sm-3"><span class="error"><?php echo $phoneError; ?></span></div>

</div>

</div>

<div class="form=group">

<div class="col-sm-5 col-sm-offset-1">

<button type="submit" title="Save Registration" class="ButtonFont">

<span class="glyphicon glyphicon-cloud-upload"></span>

</button>

</div>

<div class="col-sm-5">

<button type="reset" title="Reset Form Fields" class="ButtonFont">

<span class="glyphicon glyphicon-refresh"></span>

</button>

</div>

</div>

<div class="col-sm-5 col-sm-offset-1">

<p class="ButtonText">Save</p>

</div>

<div class="col-sm-5">

<p class="ButtonText">Reset</p>

</div>

</form>

</div>

<?php require\_once 'footer.php';?>

</body>

</html>

**Experience**

I had a positive experience during the creation of the MetricuL8 system. I often needed to think through ways of creating solutions using the client, server, and database. As I learned new PHP and MySQL techniques, I wanted to go back and refactor. Learning to retain the user choices in each ‘select’ box and apply them to a new incoming query result was a satisfying educational experience! I don’t know if I did it the best way, but I was able to use the PHP language to solve a problem. One of the key advances from the first edition is the use of stored procedures in MySQL. Some of the operations, such as adding classes, took a few queries to solve, and sometimes the web server and database server are not local to each other. In those cases, stored procedures are the better choice, because there is less back-and-forth across the network.

I would implement unit testing for the PHP and database procedure components if the project timeline were extended. PHP embedded in a functional web page is challenging to test independently, and there are several proposed frameworks (Enderlin et al., 2013). Also, a security policy should be applied to the code. Several web pages accept input as free text and place the text in SQL statements, which is a vulnerability to SQL injection attacks (Bedeković et al., 2022). The system would benefit from a central parameter filter function to be used by all functions that place input in SQL statements.

**Conclusion**

We hope you find that all functionality meets or exceeds your expectations. We generated the components, integrated the subsystems, and verified each function from the Software Requirement Specification (Tsui et al., 2018). Let us know how the user acceptance testing proceeds.

**References**

Enderlin, I., Giorgetti, A., & Bouquet, F. (2013). A Constraint Solver for PHP Arrays. 2013 IEEE Sixth International Conference on Software Testing, Verification and Validation Workshops, Software Testing, Verification and Validation Workshops (ICSTW), 2013 IEEE Sixth International Conference on, Software Testing Verification and Validation Workshop, IEEE International Conference On, 218–223. <https://doi.org/10.1109/ICSTW.2013.80>

Bedeković, N., Havaš, L., Horvat, T., & Crčić, D. (2022). The Importance of Developing Preventive Techniques for SQL Injection Attacks. Technical Journal / Tehnicki Glasnik, 16(4), 523–529. <https://doi.org/10.31803/tg-20211203090618>

Tsui, F., Karam, O., & Bernal, B. (2018). Essentials of software engineering (4th ed.). Jones & Bartlett Learning.